Compuverde
Software-defined storage

Compuverde

Technical Overview

Version 1.12
May 3, 2017

Abstract

This paper provides a detailed look at the architecture and components of the Compuverde storage system, functionalities and benefits. It covers the whole range of Compuverde scale out storage products and storage modes: Scale out NAS, Hyper Converged, Metro Cluster, Hybrid Cloud, All-Flash – and goes into the details of the supported protocols and techniques that ensure the levels of compatibility, reliability, performance, scalability, and security required to meet the highest possible telecom grade standards.

Compuverde is an established provider of Big Data and storage solutions for enterprises, service providers and telecommunications companies. Compuverde offers solutions that combine telecom-grade reliability and highly-scalable cloud-based object storage, enabling the use of environmentally friendly hardware consuming less energy. Teamed with a top-ranked university and well-known partners within the telecom and IT industry, Compuverde is creating the future of storage solutions. You can read more at www.compuverde.com.

Contents

Introduction

Compuverde recognizes the necessity for enterprises to evolve from a fixed and rigid approach to storage, to one that is flexible and pragmatic. That is why Compuverde is delivering a fully software-defined storage solution that is completely hardware-agnostic and massively scalable, thus eliminating the cost and worry of future data migrations and hardware replacements.

Compuverde is not only scale out storage, it integrates a unified file system, object store and block storage in one package. The solution is defined as a cluster of nodes with a single file system spread over all nodes. Because it is software-defined, the nodes can be built from essentially any hardware. Each node adds access points, cache, storage capacity and performance to the cluster.

Compuverde's architecture is balanced by definition, as all nodes play an equal role in the storage cluster. There is no master node, no single metadata server, lock manager or dedicated gateway node. Users and applications can access the cluster through any node, spreading the load and efficiently eliminating bottlenecks. If your storage needs to grow or latency becomes an issue, new nodes can be added to the cluster and they will immediately take a load of the compute and storage effort. This ensures that what works today will continue to work in the future.

As all nodes are self-sustaining, self-balancing and true peers, the total performance increases linearly with every node added. The result is a high performing high-availability cluster with failover functionality for service and content.

Each product offered by Compuverde covers the need for scale-out storage in a specific area:

- **Converged / Scale out NAS** is a product suitable for "bare-metal", self-sustained storage node computers running the Compuverde software.
- **Hyper Converged** allows the Compuverde solution to be installed on virtual machines running on hypervisors like VMware, Hyper-V, KVM or Xen, thus bringing the computing and the storage layers together, in the same physical machine (the hypervisor) and maximizing efficiency and performance.
- **Metro Cluster** provides complete redundancy and creates a truly high-availability cluster for mission-critical data and applications. With Metro, your data is synchronously mirrored and distributed to two independent physical locations.
- **Hybrid Cloud** is a solution for globally distributed data that provides a unified view of the file system between the local network and the public cloud. This is a key feature since such environments do not normally permit a common view of the file system that spans both public and private cloud and, as a result, the virtual machines stored in the public cloud cannot easily access files stored in the private cloud and vice versa.

With **All-Flash** storage mode, you can discard the cache disk layer and write directly to all-flash SSD storage for maximum performance, still making use of Compuverde's unique RAM cache to smooth out multiple write operations.

Key features

- Ready to use "plug and play" scale-out storage for virtualization and cloud
- Extreme performance and scalability
- Hardware agnostic
- Fully flash compliant
- Highly compatible with well-known protocols
- Telecom graded solution

Maintenance is an important part in any product's life-cycle, so it is worth noting that upgrades of the Compuverde software are done over the network, fully transparent to the clients and without

service interruptions. During an upgrade, the system can take down one node at a time, pass the IP address temporarily to another node (by using the virtual IP feature), do the upgrade and then bring the node back online before moving to the next node.

System overview

Storage requirements today are often in the range of tera-, peta- or exabytes of data, and tend to grow exponentially. Compuverde's response to these needs is a cluster of storage nodes, completely hardware agnostic, fully scalable and easily configurable through a Management Tool.

A simple setup could very well start with as few as four nodes, installed on existing or new off-the-shelf hardware, and expand linearly as demands require.

Nodes

A node is a computer with its own CPU, RAM, one or more hard disk drives for storage, and for cache: a separate disk, SSD, or preferably NVMe. The node can be either "bare metal" or a virtual machine inside a hypervisor for a hyper-converged solution.

Nodes can consist of any hardware and provide different ratios of capacity and throughput that add to the total of the cluster to which they belong.

The Compuverde software is easily installed on each node. Everything is integrated, so the customer does not need to pre-install an operating system in order to make a clean installation.

Cluster

A cluster is a collection of nodes that work together for common computing and storage purposes. The file system spans across all nodes in the cluster. All the nodes in the cluster have identical roles, which eliminates performance bottlenecks. Adding a new node to the cluster means adding its IOPS ability, cache and storage capacities to the total, resulting in reduced latency and a better user experience.

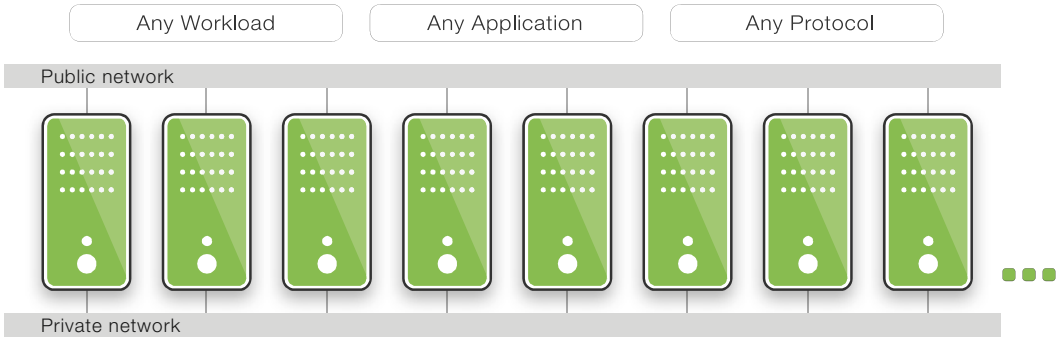

Figure 1: Compuverde cluster setup example

The more nodes in a cluster, the better: the cluster will over-all – and over time – be more balanced, both on capacity and performance levels. More nodes means that the system has more options to choose from for rebalancing the load and during the self-healing process in case one node has gone down. More nodes will also allow for more efficient erasure coding for higher redundancy, lower footprint, or both. Please refer to the section Erasure Coding for details.

"Split-brain" is a condition not welcome in storage clusters. It is a potential outcome when two parts of a cluster lose connection with each other and both parts continue to serve reads and writes, unaware that the other part is doing the same, which would lead to inconsistency. This is not an issue in a Compuverde storage cluster because at least one part of such a split cluster would

immediately set itself in a *No Cluster Agreement* state and stop operation to avoid damage. Traffic would then be directed to the part of the cluster still operational. Furthermore, the system is always looking for majority, or “quorum consensus” – for instance, the file system will require two out of three copies of an envelope (a small collection of metadata describing a folder, and the files and folders within) in order to make changes. When it comes to Hybrid Cloud, only one data center can be the owner of a given file at a time, so that if the connection between two data centers is lost, then the other data center cannot obtain ownership to make changes – all in order to keep data consistent.

Hardware can be added to the cluster from any vendor. Each piece of hardware adds to the cluster's capacity and performance accordingly. The redundancy lies within the design of the software itself, and due to the over-all symmetry of the architecture with no costly bottlenecks, reliability and performance of each single piece of hardware becomes less of an issue. This makes it possible to choose environment-friendly “green” hardware. Hardware traditionally considered to be "low-performance" is compensated by the number of nodes working together.

Network

A node should preferably have two high-speed IP network connections; one used for node-to-node communication (private) and the other to act as one of the access points to the cluster (public). For more information, please refer to the System Requirements section.

In case of malfunction to one access point, the client will be seamlessly directed to another available access point by the use of Virtual IP. For more information, refer to the section Virtual IP. Further, a load-balancing switch can be used to automatically spread the access to the entire storage cluster.

There should be at least one switch for each front- and back-end networks. Each node can utilize more than one network interface for each network, teamed up to add redundancy and to increase the performance for either node-to-node communication within the cluster, to the public network, or both. This technique is known as link aggregation or Ethernet bonding, link bundling or NIC teaming. The option is available through the advanced settings of the Management Tool and can be added at any time to increase performance, or redundancy in case one of the links should fail. When several network interfaces are bonded together within the same node, the system will look at each bonded group as one single network interface that allows for higher speed than it normally would.

Note that the storage cluster will use multicast messages when direct communication from one node to another is not practical, for instance for heartbeats or when asking which nodes are best suited for storage. For this reason, the affected network switches should support IGMP. All switches should have IGMP Snooping enabled, with only one switch in the group having both IGMP Querier and IGMP Snooping enabled.

Cache

Cache is used to dramatically reduce latency. All caches throughout the cluster are always synchronized to ensure that every node has knowledge about which node owns a copy of a certain object so that a subsequent read or write can be redirected to the appropriate node, which is faster than accessing the storage disks.

The first level of cache is RAM. This is volatile storage, so write operations are never confirmed as safe until data has been written to non-volatile SSD or disk. Consecutive reads can be almost immediate when data is found in RAM, for any client, either from a previous write or a read.

SSD, NVMe or other fast, non-volatile storage acts as the next level of cache (except for All-Flash storage mode that will skip this step due to using SSDs in the storage layer). Since this layer is normally much faster than storage, it nearly eliminates latency. When possible, very small writes and updates will be accumulated before being stored to SSD. Changes in cache are secured by using cache replication, which ensures that the changes are found on at least two nodes before being saved to persistent storage and replicated further.

As the cache is shared throughout the cluster, it is seen by the client as a cache pool having the total size of all caches on all nodes combined. For example, when a read command comes in to a node, the node can find that another node has the required data in the cache, ask for it and deliver it to the client. This process is faster than if the first node would have read the required data from spinning disks.

SSDs can be extremely fast, but because SSDs can be more delicate when it comes to repeated writing, the Compuverde software will try to collect multiple updates in blocks before writing to cache. The SSD controller firmware will make sure through wear leveling techniques that multiple writes to the same logical block will instead be made to less-used space to avoid unnecessary wear. The same cache disk is used for both read and write, which reduces the amount of operations to SSD in cases where, for instance, a file is first written and then later read by the same or another client.

Layered software approach

The Compuverde software is divided in three layers, for better performance and reliability. The entry point to the system is the **Protocol layer**, which ensures the communication between the client and the system through the protocol of choice (for example SMB, NFS or iSCSI). The **Gateway with Cache layer** follows, where the main system logics reside. Next, the **Object Store layer** is where the data is actually replicated, distributed and stored. This layer only sees data as objects and has no information about file system or structure. The software layers are explained in more detail in the Software overview section.

Virtual File System

The Virtual File System is the common file system created so that multiple clients can interact with the same stored data, regardless of which implemented protocols they are using.

The file system has been specially designed for scalability; it consists of small collections of metadata called *envelopes*. Each envelope is associated with a folder and holds information about subfolders and files belonging to that folder. Instead of being centrally stored in a Gateway node or a database, the envelopes are stored as objects in the Object Store, just as any other regular data. This creates a robust architecture that allows nearly infinite expansion. While being present in the cache, the envelopes can be mirrored to two copies in two different caches, just like content data. In the Object Store layer, they are mirrored to a minimum of three locations.

Each item in an envelope (i.e. each item in a folder) has one or more globally unique identifiers (GUID) – since all files are organized in collections of data called extents, with each extent having a size of maximum 1 GB – their own GUID and an offset value. The offset is useful when data is added with an arbitrary gap from the previous part; having an offset value means that there is no need to store the gap itself. The envelope also contains information about item type, time stamps, snapshots, item size and a multicast IP address – which is used when a node wants to ask all other nodes if they have relevant information about the item.

There is literally no physical limit to this structure, which allows for millions of files and folders within each folder and billions of files in total. File names use UTF-8 for compatibility and each file name or folder name can be up to 260 characters in length.

To ensure consistency, every change to the file system is done through transactions so that the file system cannot be left in an intermediate state in case of malfunction.

Management Tool

The Management Tool can be installed on any Windows PC connected to the storage cluster's private network. The Tool is not part of the cluster or of any node-to-node communication, it is merely a means to perform administrative tasks, like monitor the system, change configuration features, access logs, or roll out new firmware updates, all through a self-explanatory GUI.

Updated configuration settings are automatically distributed throughout the storage cluster. The Management Tool comes with a user guide that details each feature and option.

The Management Tool will soon also be available as a plug-in to the VMware vSphere Web Client for use with the Hyper-Converged product.

Products

The Compuverde solution can be set up in four distinct ways, defining four scale-out storage products: Scale out NAS, Hyper-Converged, Metro Cluster and Hybrid Cloud. Regardless of which product you choose, the software installation and cluster setup are done in a few simple steps:

1. Install the Compuverde product software on each node and give a name to the node
2. Connect each node to the public (and private) network(s)
3. Install the Compuverde Management Tool on a Windows client that can access the private network used by the nodes
4. In the Management Tool, run the wizard to create a cluster and add each node to it
5. Run the file system wizard to create a file system on the new cluster

Scale out NAS

In this setup, each node is a self-sustained computer with Compuverde software installed, complete with a set of predefined protocols. Each computer is equipped with one or more disks for cache and a number of hard drives for storage (or, for All-Flash, only SSD for storage). The figure below is a conceptual view of three nodes. The system accepts interaction from applications and users through Ethernet and is keeping itself internally synchronized, horizontally throughout the

cluster and vertically down to the storage. The file system is spanning over all nodes. Each node sees the same complete file system and acts as a file server towards the client accessing it.

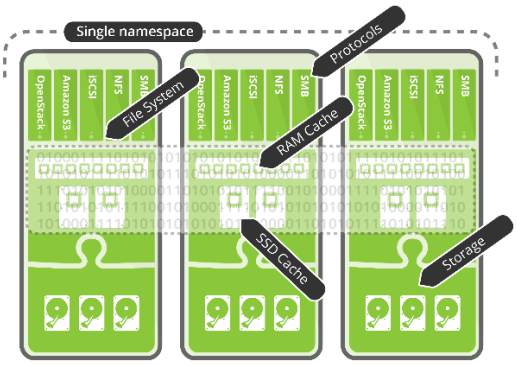

Figure 2: Schematic view of 3 nodes, scale out storage

Note: Though it is technically possible to make a cluster out of only three nodes, it is highly recommended to use more, to allow headroom for the self-healing and the balancing processes to work properly. Four is a recommended minimum.

Hyper Converged

Compuverde can be installed on virtual machines provided by hypervisors such as VMware, Microsoft Hyper-V, KVM or Xen. This allows the user to create own virtual machines and place them on the Compuverde storage cluster, which, in turn, is physically mapped to the storage resources of the hypervisor.

For example, three virtual machines with the Compuverde software installed could be running on three VMware ESXi hosts (one Compuverde VM per ESXi host). The Compuverde VMs would use physical storage provided by the respective ESXi hosts in order to build the Compuverde cluster, which spans across all ESXi hosts. Then, on any ESXi host, a user could create any number of new regular VMs running Windows or Linux and place them on the Compuverde storage. Moreover, these user virtual machines can share the Compuverde storage for common user data. At the same time, the storage can be shared to users outside the hypervisor.

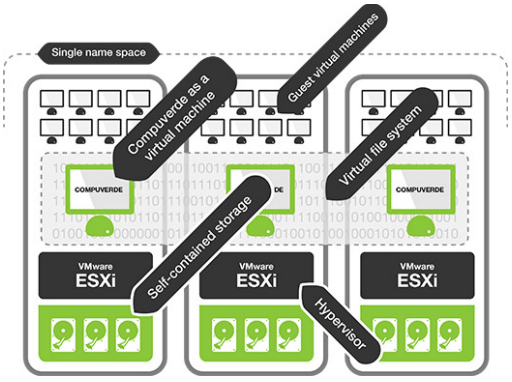

Figure 3: Schematic view of three nodes in hyper-converged setup

For performance, each node is mapped so that each virtual machine will use the access point locally on that same hypervisor, thus also accessing the local cache directly. Considering that the virtual machines are spread through the cluster, this will also balance the workload significantly. Potential problems caused by boot storms and similar storage issues are now eliminated because the load is evenly spread across the cluster and the cache is utilized to serve all the virtual machines directly.

Another advantage of this approach is that the same hardware is utilized for computing, virtualization and storage. To not drain system resources, Compuverde intelligently allocates only parts of free and available RAM.

Metro Cluster

Metro Cluster from Compuverde provides complete redundancy and creates a truly high-availability cluster for mission-critical data and applications. With Metro, your data is synchronously mirrored and distributed to two independent physical locations.

Whether you choose Compuverde scale-out NAS or a hyper converged virtualized setup, Metro will mirror and distribute all data and changes to both parts of the cluster in real-time. You will be able to use the nodes both from your primary and secondary part of the cluster to access the storage and all nodes will have the same view of the unified files system.

In case one location should suddenly go down due to failure caused by hardware, network or other disaster, data is kept safe on the other location by use of erasure coding within the sub-cluster. The delivery of service can continue practically uninterrupted, but only in situations where data integrity is not at risk.

Hybrid Cloud

Compuverde Hybrid Cloud is the solution for globally distributed data. Clusters of nodes can be spread on several locations and interconnected through the Internet. For performance, the user is routed to the nearest data center. The data centers synchronize in the background.

Figure 4: Hybrid Cloud

Extensive use of cache and synchronization of metadata make sure that the data center has the same view of the data as all the other data centers in the cluster. Should, for instance, a data center in one part of the world suddenly lose connection, Compuverde's solution allows for all communication to be seamlessly migrated to the next most suitable datacenter in the cluster so that the data provider can continue to deliver a positive user experience. Should you add yet another location and datacenter to your setup, all will immediately make use of the features and benefits that the solution offers. Using the solution also has the effect that ownership of data can move between different data centers depending on the last location for execution.

All-Flash

Compuverde recognizes the importance of a fast cache layer for enterprise storage to dramatically speed up the storage solution. However, the size of the cache layer can become a limitation for some continuously demanding use-cases. For this reason, the Compuverde All-flash solution discards the cache disk layer altogether, still making use of the RAM cache, and writes directly to your all-flash object store.

An SSD storage solution aids to your power savings by operating with less energy consumption compared to mechanical drives. As with all Compuverde storage, the use of multitenancy that allows for you to deploy multiple, independent file systems on the same storage infrastructure, further reduces your investment costs.

The all flash storage mode can be combined with Scale out NAS, Hyper converged and Metro.

Quality attributes

Compatible

Compuverde provides access to the storage through various protocols, depending on the needs of the clients. If so desired, different clients using different protocols can access the same data simultaneously, on any gateway (access point), so that the choice of protocol is completely up to the client or the application. It is also possible to create multiple domains and multiple file systems within the same storage cluster, so called multi-tenancy – simplifying the administration compared to having separate storage for each domain. Protocols include SMB, NFS, iSCSI, OpenStack Swift and Amazon S3. For more details, please refer to the Supported protocols section.

The view of the file system through each node is strictly consistent, so that any modification on one gateway node is instantly available from any other gateway node. This is ensured by metadata being synchronized. A gateway node cannot deliver data until it is confirmed by the storage cluster that it is the latest version (or an earlier version if a specific snapshot was requested).

Reliable

Reliability lies at the core of Compuverde, designed to provide telecom-grade availability of 99.999 percent or better, also called "five nines". This is achieved due to the symmetry of the architecture and by keeping the system core small, clean and efficient.

Rolling upgrades, combined with the Virtual IP feature, enable you to upgrade the system on the fly without downtime: nodes are taken offline one at a time for upgrade, the next node will automatically obtain that node's IP and the rest of the cluster will continue to serve uninterrupted, totally seamless from the view of the client.

Compuverde uses erasure coding for data redundancy and protection. Data is striped across nodes and locations, not simply across disks as with traditional RAID. In case of hardware failure, each remaining node will be notified and start to recreate the missing data. Since all remaining nodes are working in parallel to recreate the missing data, the storage cluster is quickly returned to a normal state and is ready to handle any upcoming hardware problems. This is crucial when dealing with large hard disk drives and volumes that would otherwise require a long time to fully rebuild.

The more nodes that are added to a storage cluster, the faster the recreation process of missing data goes – despite the now larger capacity – because there will be a relatively smaller amount of missing data for each node to recreate.

High performing

Scaling up by adding more storage nodes provides a linear increase in average performance in the cluster. All the resources in the storage cluster are aggregated, like CPU, bandwidth, storage and cache pool, giving a persistent high throughput for a high number of simultaneous users. Because all the nodes in the cluster also share the work of recreating lost data, the time frame from a hardware failure until the cluster is fully restored is dramatically shortened, with minimal impact on

the system performance. The more nodes in the cluster, the less time it takes to complete the tasks.

Scalable

Every element of the architecture is designed for scalability – for instance, the file system and metadata needed by the gateways to form structured storage (SMB/NFS) within an unstructured cluster. Because the metadata is crucial information that should be protected from loss, it is best stored as objects within the storage cluster rather than in the gateways, as seen elsewhere. Internally, each file (and each block of data) is identified with a 128-bit GUID. The metadata is cached, synchronized, duplicated and stored using the same algorithms as normal files. This not only creates a more robust and less complex architecture, but it also allows for extreme scalability up to hundreds of petabytes and billions of files for each storage cluster.

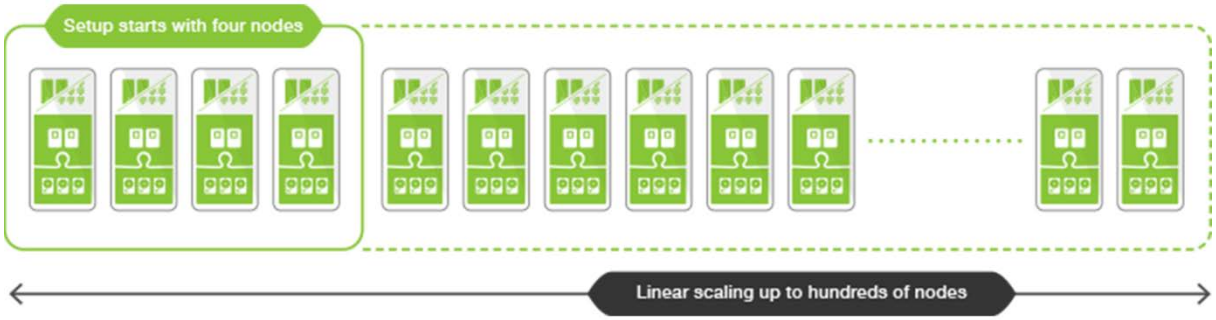

Figure 5: Compuverde Linear Scalability

There is no built-in limitation in terms of number of nodes in a single cluster. Hundreds of storage nodes can be added to the same cluster. Separate storage clusters can be combined, allowing exabytes of data.

Secure

Data security is provided through redundancy and authentication, as a layer of security can be introduced by verifying the credentials of the users or applications before allowing access to the data. Read more about authentication in the section Authentication.

Features

Multitenancy

Compuverde allows multiple, separate domains and file systems spanning over one single storage cluster, reducing costs by simplifying the administration and allowing available storage resources to be used more efficiently – as opposed to having separate storage for each domain. Each domain and file system are separated, using their own IP addresses for access, authentication mechanisms and set of protocols, still located on the same hardware. This way, the costs can be reduced as the overhead needed for the storage solution, in terms of CPU, infrastructure and storage capacity, can be utilized for all the domains combined.

Disk Quota

A disk quota is a limit that can be set in the Management Tool, which restricts the use of storage space for a specific domain, file system, folder or sub-folder. The limit is specified in number of GB, TB or PB and will then apply for all users with access to the folder or share. The quota does not allocate storage, which would require an amount of overhead and less than optimal use of

resources in cases where a number of quotas are set and each quota is not fully utilized. This means that, if the administrator chooses, the sum of all the quotas on one hierarchical level may exceed the quota or limit for the parent level or the storage itself.

When a write operation tries to write beyond the given limit, the operation will be aborted and the client will be notified that the write was not successful.

Virtual IP

Virtual IP is a failover mechanism to make sure that all the nodes in a cluster appear available at all times, continuing to deliver the service when a node is taken down for upgrade or in the event of failure. This is done by moving the IP address of the node going offline to another node, which becomes responsible for the new IP address in addition to its own. The process is automatic and fast. The cost is a temporary decrease in performance, until the affected node is ready to step back in and ease the load.

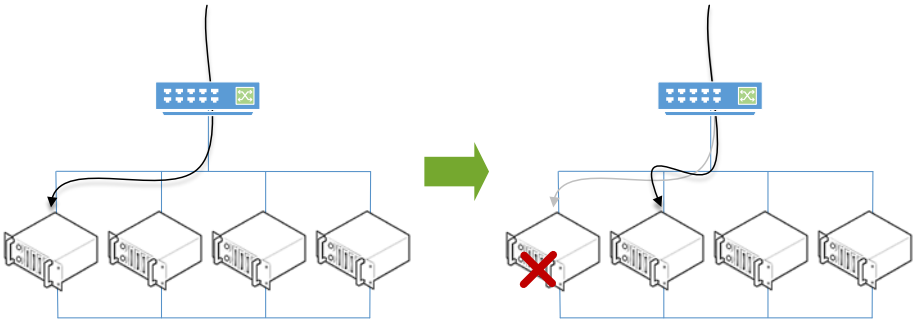

Figure 6: Virtual IP acquired by another node

When a node is taken offline in a controlled manner, the node will select another node in the cluster to take over its IP address and the transfer will be done transparently and seamlessly to the clients.

When a node goes down due to a failure, the cluster will detect the failure and then assign the missing IP address to another node. In this way, the IP address would still be available from the outside as if nothing happened, while the cluster will continue to operate with the remaining nodes.

Separate IP for public and private network is required, and this can be accomplished even on nodes with single network interfaces. Virtual IP is enabled through the Compuverde Management Tool.

Auto Rebalancing

The nodes in the storage cluster will always aim to stay balanced, i.e. to utilize an equal amount of storage and IOPS compared to each node's total abilities. When you add a new blank node to the cluster, this new node would then take a disproportionately heavy load of new data written to it. The Auto Rebalancing feature remedies this by moving some data to or from any node found to be out of balance, thus avoiding hot spots completely. As soon as the new node is included in the cluster, the pre-existing nodes are redistributing parts of their stored data to the new node. The feature will generate some additional data traffic on the private network and therefore the feature is optional.

This is an essential measure because the new node could otherwise end up being the only node to receive new data, which – as new data has a higher probability of being read back soon compared to older data – would lead to a hotspot. The cluster stays balanced and all new data will continue to be evenly distributed.

Rolling Upgrade

Compuverde supports rolling upgrades over the network so that the software can be updated without taking the system offline. This is done through the Management Tool. By using Virtual IP on the nodes, each node can be taken offline and another node will seamlessly take its place, now responding to both its own and the offline node's IP address. After the upgrade, the node will join the cluster and reclaim its IP address so that the next node can be taken offline for upgrade. This way, the cluster continues to deliver the service uninterrupted during the upgrade process.

File Policy

File policy is a feature to make automatic actions to files and content, for instance when they reach a certain age. It could be to move the data to another storage tier, to set another file encoding, to set WORM or to simply remove old data. Using the Management Tool, a file policy can be set to a folder at any hierarchical level, and will then apply to all files in all sub-folders below.

A file policy can be triggered by the age of the file (i.e. the time interval since the file was last modified: days, weeks, months or years). The files are chosen by file name pattern (e.g. "*.jpg", single and multiple character wildcards are allowed). When no age is specified, the rules apply continuously. A new file encoding (e.g. Erasure 2+1) can be specified for files that are moved or re-written. If the intention is to erase files after a set time interval, since the file was last modified, an optional retention period can be set. Another option is to have the files made WORM (Write Once Read Many) for archival. The files will then be set read-only and cannot be edited or deleted until changed to allow write.

Note that file policy feature does not move or change the location of items within the file system. Instead, it moves or rewrites the data to another physical location on the storage (e.g. another tier).

Snapshot

Snapshots give you the ability to retrieve earlier versions of files or folders in case of unwanted changes or deletion of files or folders. Snapshots can be configured so that one is made for example every hour, every day or on a particular day of the week.

Logically, a snapshot is similar to a copy of the underlying file structure and all its files and content. However, technically, to save space and to avoid loss in performance – and, most importantly, to ensure that all the folders and files in the snapshot are made at the exact same moment in time – the snapshot is merely a time stamp with information that the snapshot has been performed. The first time a change is being performed to a file after a snapshot has been taken, the part of the file that is going to be modified is copied and marked with the time stamp of the snapshot, before the file is actually modified. This method is known as copy-on-write.

Each update generates copies in block sizes as small as 128 kB. New updates within the same 128 kB block will still result, if no additional snapshot is taken in the meantime, in only one copy of the modified block.

The file system will seamlessly deliver files from each snapshot, when requested. There can be up to 253 snapshots for each directory. As the snapshot is just a pointer to existing data and the data is stored on the same hardware, it should only be used as intended – never as a backup.

Once a snapshot policy is assigned to a folder, it will be valid for all sub-folders and files. Thus, there cannot be another snapshot policy on sub-folders. In order to assign a snapshot policy to a sub-folder, the first snapshot policy must be removed and then the new one applied on sub-folders.

Queuing and prioritization

The self-healing processes use separate queues to prioritize the tasks that need to be performed in order to keep the storage cluster healthy and consistent. For instance, protection and replication are queued with higher prioritization than the removing of excessive data.

Separated mode

Compuverde separated mode is a customization that can be applied to any product in order to meet extreme demands. In this mode, there are two types of nodes: Gateway nodes with Cache and Object Store nodes for storage. For all other configurations, everything is integrated into each node.

In this mode, the gateway layer and storage layer run separately, allowing the data centers the flexibility to handle any extreme demands, whether it is for capacity, for performance, or both.

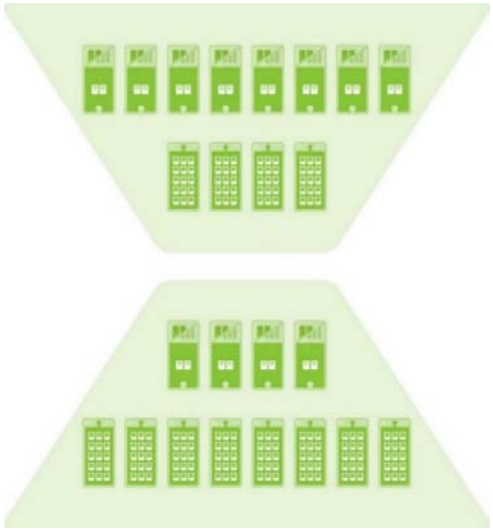

Figure 7: Schematic view of separated mode – scaled for performance (top) or capacity (below)

The configuration is very flexible and can be used in a setup to meet extreme storage needs. The following figure illustrates the way performance and capacity can be scaled separately by adding storage nodes or gateway nodes respectively, compared to traditional solutions.

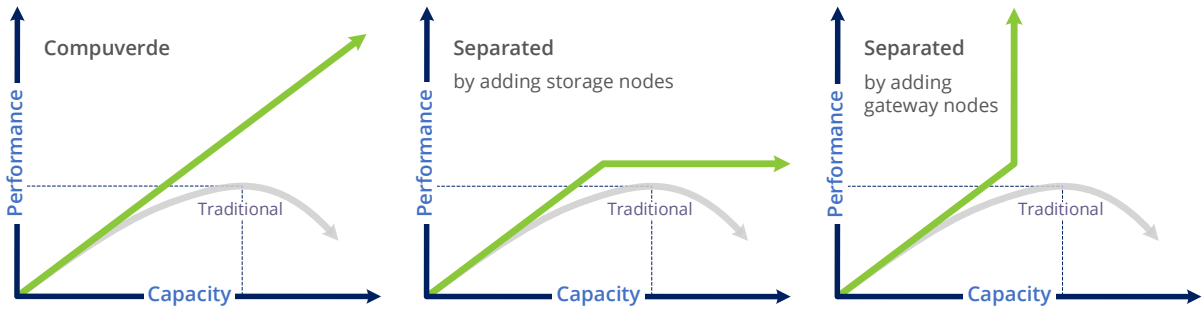

Figure 8: Flexibility of Compuverde separated mode

Data protection

Data Integrity

Compuverde uses a variety of methods and techniques to ensure data integrity even in the most stressful working conditions, like heavy multi-concurrent IO activity to the storage, sudden power loss or hardware failure.

- Intelligent locking system allows multiple clients to do concurrent read and write to the same files at the same time, by protecting the data at the required level and byte range.
- No intermediate state in case of power loss or malfunction. Changes are confirmed only when stored on persistent storage.
- No "split brain" condition in case of hardware failure or lost connection; Compuverde requires majority (quorum consensus) for changes to be performed.
- No hash collision, due to truly unique 124-bit ID for each block of data, calculated from previous block ID.
- Self-healing storage cluster – all nodes in the cluster are responsible for monitoring, replicating and seamlessly recreating data.

Power loss

Regular operations on the storage, like storing and reading data, are secured by the balanced architecture with no single point of entry, i.e. no single point of failure, and by the storage redundancy, e.g. Erasure Coding. This is covered in the section Erasure Coding.

What could happen to a storage system in case of power loss is that some files might be in the process of being written to storage, or to cache, and therefore be lost in the process, or the file system may be left in an invalid intermediate state. These potential risks are solved by using transactions, which means that changes are either stored and confirmed, or they are not – there is no intermediate state. Also, there is an option to use write cache replication for added protection, meaning that the cache data is replicated to another cache on another node before being written to persistent storage.

When a client initiates a write operation on a file, the data is received and buffered. The client gets the confirmation that the data is received but not yet safely stored. A transaction will start updating the file system. If this, for any reason, should fail, e.g. due to power loss, then no change has been made and thus no final confirmation was given to the client. With cache replication enabled, the transaction will just continue on the replicated cache elsewhere and then be confirmed. We will only confirm the update after this process is successful. A version number is added to each piece of data to ensure that any consecutive read, from anywhere in the cluster, will receive the correct version of the file.

If a failing node comes alive, all non-volatile cache will be checked and saved to disk only if no other node has already saved the data and if no newer version has been committed in the meantime, using version number for each block for verification.

Metadata is replicated in the cache in the same way as content data. As metadata is relatively small in size compared to the content, it is always mirrored to at least three locations in the Object Store layer, instead of being erasure coded.

Hardware failure

A service named Cluster Monitor is running in the Object Store layer of each node, monitoring its own health and verifying through heartbeats that all other nodes in the cluster are alive. In the event of a hardware failure, power failure, NIC malfunction etc., the node is declared dead. Each and every node will automatically start replicating its own data (the parts that were located on the dead node) onto available storage throughout the cluster, thus keeping the vulnerability window to a minimum. Then, when a disk is replaced or a node is reinserted, no rebuilding of an entire volume will be necessary as would be the case for RAID disks, eliminating the chance of a second failure during a time-consuming rebuild.

The cluster resilience limit is the total number of nodes that can be lost without losing any data. This feature can be set on cluster level in the Management Tool. If the limit is exceeded – e.g. a second node goes down on a cluster where the resilience limit is set to 1 – then the cluster will close down to avoid further damage to existing data.

For details on erasure coding, see the section Erasure Coding.

Self-healing storage cluster

The Object Store layer is responsible for erasure coding and redundancy. An ongoing self-healing process is constantly scanning for consistency issues. Missing parts are queued and re-created as fast as possible, while excessive parts are scheduled for removal with a lower priority. In case of failure to nodes, there is no need for “hot spare parts”. Instead, each node will start replicating the affected files to available free space on the cluster. The administrator will be alerted so that the broken node can be serviced and reinserted on the fly.

A storage cluster can be in one of three possible states:

- Healthy cluster – All the storage nodes and disks in the cluster are online and operational
- Degraded cluster – One or more storage nodes or disks are offline
- No cluster agreement – The number of nodes online is not sufficient for normal functioning

A degraded cluster will start replication using available storage in order to secure the data.

The system automatically checks the cluster and content in several ways to correct any potential error and to maintain security and effectiveness, including:

- Cluster Monitor / Heartbeats
- Search Cluster
- Read, Write and Update

Cluster Monitor

Detection of lost nodes is done through heartbeats. A multicast message is sent through the cluster and received by all nodes. When a node has not sent any such message for a predefined number of seconds, it is declared dead by the cluster. All the other nodes will start the Search Cluster operation in order to rebuild the content of the missing node.

Search Cluster

When the cluster detects that a node is offline, either by missing heartbeats or when a node is taken offline by the administrator, each node simultaneously starts a local search for all files that were shared with the offline node, and immediately queue the results for a replication using available storage on the cluster, ensuring that the data is still protected. This is possible because each node has information about all the locations of any file found on the node that went offline. A grace period can be set so that when you intentionally take a node offline, the cluster will not immediately start replication activities.

The process needs to be fast because we do not want to leave the system in a less protected state for too long; i.e. the vulnerability window should be kept narrow. The process is extremely fast because each node is responsible for its own files, making it into a many-to-many replication with no central orchestration. Tests have shown that 24 moderately sized nodes were able to replicate 5 million unstructured objects, each of 1 MB in size, in only 19 minutes. With other replicating systems, a task like this would be measured in hours or days.

When the "declared dead" node later comes alive, after it has checked its own cache and made itself available to the cluster, the replication process will abort and another process will start to

remove the now-excessive (old) chunks of data. This is a separate queue with lower priority due to the fact that excessive data is less critical. Every chunk of data is equipped with a version number so that the latest version is kept consistent.

Cluster health, status and replication can be monitored in real time through the Management Tool.

Read, Write and Update

Another part of the self-healing process is each ordinary read, write and update. Each time a file is accessed on the lower Object Store layer, a list with metadata is generated from all the nodes that store the file. All the lists from all the nodes are compared to ensure that all nodes are consistent. If any inconsistency is found, the issue will be queued for repair.

Erasure coding

A common requirement for a storage cluster is security through redundancy. The most elementary way is by copying or mirroring, and while this is an option for very small amounts of data, like metadata, the downside becomes clear for big data: three copies of data required to allow any two nodes in a storage cluster of any size to break down, would lead to a footprint of 300 percent, meaning that two thirds are for redundancy. This goes for any storage size and any amount of nodes.

By using erasure coding, the footprint is reduced significantly. Each block of data is sliced into smaller pieces and sent to a number of storage nodes according to the chosen erasure coding. For example, with erasure coding 2+1, each block of data is sliced into two pieces, one piece is added for redundancy, and then sent to three different nodes. Three nodes are chosen for each block of data so that all the nodes in the cluster will receive pieces according to each node's individual capacity and location.

For a 8+2 setup, the space required for the redundancy is only 25 percent – a reduction of 87 percent compared to plain mirroring. Still, any two nodes are allowed to fail, whether these two nodes contain the original data or the redundancy, or any combination of both. The storage will seamlessly recreate the original data from any combination of remaining slices.

Erasure coding is similar to RAID 5 and 6, but as Compuverde intelligently spreads the slices across nodes and locations, data and services stay protected and sustained in case of failure to not only disk drives, but entire nodes.

There are no drawbacks to erasure coding other than the calculations required when doing write or update operations. However, the client does not experience any additional latency, since all commands are considered completed when confirmed by the non-volatile Cache layer; then the Object Store layer makes sure the erasure coding is applied in the background, before the data is distributed and saved to storage. Note that for performance reasons, the cache is using optional mirroring for metadata, not erasure coding.

When choosing erasure coding for a file system, it is relevant to consider the required redundancy, storage efficiency (footprint) and CPU load. Higher levels of redundancy require more calculations for each block to be written. The cache will keep the latency down for read and write operations. There are normally no extra calculations for reading files, except in the event of missing nodes.

	Minimum number of nodes	*Nodes allowed to fail*	*Footprint*	*Storage efficiency*	*Load each disk*	*CPU usage*	*Note*
2+1	4	1	150 %	67 %	50 %	Low	
3+1	5	1	133 %	75 %	33 &	Low	
4+1	6	1	125 %	80 %	25 %	Low	
5+1	7	1	120 %	83 %	20 %	Low	
6+1	8	1	117 %	86 %	17 %	Low	
8+1	10	1	113 %	89 %	13 %	Low	*
2+2	5	2	200 %	50 %	50 %	Moderate	
3+2	6	2	167 %	60 %	33 %	Moderate	
4+2	7	2	150 %	67 %	25 %	Moderate	
5+2	8	2	140 %	71 %	20 %	Moderate	
6+2	9	2	133 %	75 %	17 %	Moderate	
8+2	11	2	125 %	80 %	13 %	Moderate	*

Erasure Coding options marked with an asterisk are currently not available for Metro Cluster.

Recommended minimum number of nodes in a storage cluster is: n+k+1 e.g. for erasure coding 3+2, minimum six nodes should be used.

Compuverde uses industry-standard erasure coding:

n+1 Similar to RAID 5, n+1 uses logical XOR and has a very low impact on the CPU, even for writing. Files are easily retrieved in case one node happens to fail by using XOR "in reverse". It is a simple, fast and safe process.

n+2 Similar to RAID 6, there is a small penalty for write and update because the second slice is coded with XOR according to a table. We estimate this to be four times as intensive as n+1. Reading a file is normally as fast, safe and simple as for n+1, but the obvious advantage is that any two nodes can be missing.

When data is accessed from the storage layer (that is: when data is not present in any node's cache) then if one or two nodes are busy or for other reasons cannot respond in time, then it is possible for the storage layer (due to data redundancy) to deliver the required data without waiting for the busy nodes and thereby keeping latency to a minimum. Note that even though the erasure level is set for an entire file system, each block of data will remember its individual setting. So that if the level is altered, then the existing files will still remain unaltered until the file is rewritten. This can also be done by using file policies.

The file policy feature allows the system to automatically move older files to another tier, with a different level of redundancy if desired, for archiving or for other purposes.

Authentication

Authentication services provide a layer of security by verifying users' credentials or applications before allowing access to read or modify data. Compuverde supports the following services, methods and protocols:

- AD - Active Directory
- LDAP - Lightweight Directory Access Protocol
- NIS - Network Information Service
- NTLMv1 / NTLMv2 - Windows NT LAN Manager
- Kerberos
- Local Users & Groups

Each node in the cluster automatically shares the same configuration, making it very easy to manage. There can be multiple authentication mechanisms for each domain and file system in the same cluster.

AD – Active Directory

Active Directory is a directory service for Windows domain networks. The Active Directory domain controller stores information about network resources and security principals (users and groups). Each security principal is assigned a unique security identifier (SID). The main reason for joining the cluster to a domain controller is to perform authentication.

Kerberos provides enhanced authentication and standardization in order to cooperate with other operating systems.

Kerberos

Kerberos is a protocol for authentication. It is an integral part of Active Directory and is also used with other directory services. A challenge-response mechanism is used so that clients are able to prove their identities without sending a password to the server.

LDAP – Lightweight Directory Access Protocol

Lightweight Directory Access Protocol is an open, vendor-neutral, industry-standard protocol to enable access to directory services for authentication. Hence, LDAP can be used across many platforms. Active Directory uses LDAP for communication.

NIS – Network Information Service

Network Information Service is a directory services protocol. NIS is different from NIS+ which is not supported.

NTLM – NT LAN Manager - v1 - v2

Windows NT LAN Manager is a suite of Microsoft security protocols that provides challenge-response authentication, integrity, and confidentiality to clients.

Local Users & Groups

An alternative to having a dedicated domain server is using Local Users & Groups that can be set up through the Management Tool. This is also where the optional secret key for Amazon S3 is specified.

Supported protocols

Compuverde supports client operating systems and clients using the following protocols:

- SMB
- NFS
- iSCSI
- OpenStack Swift
- Amazon S3
- NNTP backend storage

All data below the Protocol layer, from the Gateway with Cache layer and down to the Object Store layer, is shared between the protocols so that changes made using one protocol on one gateway are instantly viewable to other systems using another protocol, even on another gateway.

Asynchronous writes – which enable more parallelism compared to synchronous writes – are supported for all protocols and may improve performance. The client can send asynchronous write

requests to the cluster, which acknowledges receiving the data before buffering and actually storing. However, the final confirmation is sent to the client only after the data has been safely stored to non-volatile cache, ensuring that the data is safe.

SMB

SMB, Server Message Block (previously known as CIFS), is the network file system and directory service mainly used by Microsoft systems. Compuverde supports all versions of SMB up to 3.0 being used in Windows 8, Windows Server 2012 and later.

Fully supported features as of January 2016, SMB versions 1 – 3

SMB 1	Basic features
	Home directory
	Authentication NTLM v1 - v2 and Kerberos
	ABE
	Offline folder
SMB 2	Crediting
	Server-side signing
	Notify: 1 level
	Oplock
	File leasing
	CopyChunk
SMB 3	SMB Multichannel
	SMB Encryption
	Signing

NFS

NFS, Network File System, is a distributed file system generally used by clients running UNIX and Linux. A client can access the file system the same way as if it were on a local file system without the client being aware of the actual location of the file. Compuverde supports NFS version 3, 4.0 and 4.1, which is the current release.

Fully supported features as of January 2016, NFS versions 3 – 4.1

NFS 3	Basic features
NFS 4	ACL
	Authentication Kerberos
	Integrity Kerberos
	Encryption Kerberos
	File locking
	Session lease
	Client recovery
NFS 4.1	Exactly Once Semantics
	Session trunking
	pNFS File layout

iSCSI

One or more iSCSI Targets can easily be added to any cluster through the Compuverde Management tool. One or more logical units, LUs, can be added and connected to by any iSCSI initiator. By using Multipath I/O (MPIO), we deliver a high-quality and reliable storage service with failover and load balancing capability. Contrary to other protocols, due to the nature of iSCSI, the

file structure one may decide to create inside an iSCSI target will not be accessible or viewable through other protocols.

Fully supported features as of January 2016

iSCSI	Multiple sessions from different initiators
	MPIO (Multipath) - The target can be reached on multiple paths (failover)
	Digests (CRC32 Error detection)
	CHAP Authentication

OpenStack Swift

OpenStack Swift is a RESTful Web protocol for a distributed object/blob store. Objects are stored in containers instead of files in directories and can be returned as XML, JSON or plain text. Data is organized using accounts, containers and objects.

Block storage is provided through an OpenStack Cinder plug-in. OpenStack Icehouse has to be installed, and there has to be an OpenStack controller node with Cinder Volume Service installed.

Fully supported features as of January 2016

Swift	Accounts (get)
	Containers (get, put, delete, post, head)
	Objects (get, put, copy, delete, head, post)
Cinder	Operations on Volumes (create, delete, extend, clone)
	Create Image from Volume and Volume from Image
	Host assisted Volume Migration
	Manage existing Volume

Amazon S3

Amazon S3, Simple Storage Service, is an API for organizing objects in buckets. Buckets and objects can be created, listed and retrieved using a REST-style HTTP interface, returned as XML. You can also host static websites by using the website feature in Amazon S3.

Fully supported features as of January 2016

Amazon S3	Common Headers
	Authenticating Requests - AWS Signature Version 2
	Operations on Buckets (list multipart, get, delete, location, head, put, acl)
	Operations on Objects (list, get, head, post, put, copy, upload part/multipart, delete, acl)
	Operations on Bucket Website (put, delete, get bucket website)

NNTP

Compuverde provides a limited command set to support NNTP backend storage, so that Usenet servers, e.g. Diablo Usenet Software, can connect to a Compuverde cluster for storage.

Supported features as of April 2017

NNTP	Upload	IHAVE, TAKETHIS
	Download	ARTICLE, HEAD, BODY
	Info	CHECK, STAT
	Delete	DELETE
	Management	HELP, CAPABILITIES, QUIT, MODE STREAM

Software overview

To ensure performance and reliability, the system software and all the protocols are implemented by Compuverde in C/C++. This is to assure a much tighter and consistent integration between the protocols and the internal file system, so that all implemented features in one protocol can be made interoperable in real time with features in any other protocol. The result is predictable high performance, stability and a smaller footprint.

For example, if one user opens a file in SMB and is granted a range lock on that file, and then another user opens the same file in NFS, then the first range lock is still valid, even if the two connections are made through different protocols or different gateways.

There is no use of third-party or open source libraries, giving the developers full control of the code, resulting in a smaller footprint and better reliability. CentOS 7 is embedded with the installation, which is a stable, predictable and robust Linux platform.

Software layers

For reliability and compatibility, the software architecture has been divided into three main logical layers:

1. Protocol layer
2. Gateway with Cache layer
3. Object Store layer

Protocol layer

The Protocol layer is what makes Compuverde compatible with a wide range of existing systems, OSs and applications. Data below the protocol layer is shared between the protocols so that changes made using one protocol or file system on one gateway are instantly viewable to other systems using another protocol or file system, even on another gateway. iSCSI on the other hand, can be viewed as a “bucket” of storage to which the client can do what it pleases. This bucket is still stored in the object store as objects, the same way as other files, but is not accessible through any other protocols than iSCSI.

The Protocol layer delivers the file system and functionality for SMB, NFS, iSCSI, OpenStack Swift and Amazon S3.

Gateway with Cache layer

The Gateway with Cache layer, as the name implies, consists of two main parts:

- Gateway
- Cache

The Gateway redirects and conducts all file operations, communicates with all other gateways in the cluster and uses the Cache for faster reads and writes. For the Cache, there is a front-end and a back-end API. The Gateway communicates with the Cache front-end and the Object Store communicates with the Cache back-end. The Cache holds information about what parts of files need to be stored in the Object Store. A background process running in the Object Store then handles these objects and makes sure they are replicated and stored on multiple nodes.

Information about all the caches are synchronized throughout the cluster so that any read or write can be redirected to the cache holding the file.

Object Store layer

The Object Store handles data as objects and has no notion about file structure. It communicates with other nodes' Object Stores through the back-end network interface in order for the replication to work seamlessly. The layer consists of four main parts:

- File Service: Localize, read, write and delete files
- Replication Service: Replication of missing parts and removal of excessive parts
- Cluster Monitor: Heartbeat and condition of all other nodes
- Management: An API for configuration and information

The File Service is responsible for saving all cached data and to retrieve non-cached data. The Replication Service will take care of data redundancy. The Cluster Monitor is responsible for monitoring the cluster through heartbeats and to scan content for any inconsistency. The Management API provides to the Management Tool information and means to update the configuration.

The self-healing and replicating processes ensure that data at all times is secured and that the Gateway will have a consistent and clean access to the storage. At the same time, the available capacity is kept balanced.

Read operations

A read request starts with a certain protocol's respective read command to the gateway of a node of choice. The node will quickly look up and determine whether the file is owned by another node. If the file is owned by another node, the first node will check if the file is found in its own cache, and, if yes, it will ask the owner node directly whether this file can be used as it is (in other words, if the respective version is the latest one, thus ensuring data consistency). If the reply is yes, then the read data can be returned immediately. Otherwise, the request is forwarded to the node that owns the file.

The node that owns the file will look for the content in its own cache, and then any other node's cache. If found, the node will return the content, in this case almost instantly.

If the data does not exist in the Cache layer, the node will ask the Object Store to retrieve it. The Object Store layer will send a multicast message asking for file slices. Each node that can provide a piece will reply directly to the node asking, minimizing multicast messages. Included in this reply is the version number for consistency, and information about the node's current load, in addition to a list of every other node known to share the same file. The node then decides what nodes to retrieve the file from. Pieces are collected, joined and returned. This whole process is seamless and invisible to the user.

When a client asks for a very small piece of data, the whole block where that piece is located is cached. If the client later wants to read more from the same block, the data is readily available in cache.

Write operations

A write or update request starts with a certain protocol's respective command sent to the gateway of one node of choice. The Gateway layer will determine whether the file that the write is addressed to is owned by another node and, if yes, it will forward the request to that node.

For protection, the data is sent to yet another node for replication, so that it will be written to two caches at the same time. Make sure "write cache replication" is enabled in the Management Tool for this replication to take place.

The content of each file is handled in blocks of up to 1 GB in size. Each block – or extent – is given a unique 128-bit GUID, calculated as a variation of the previous GUID in such a way that hash collision is not possible. In addition, each extent is given a version number and an offset value that indicates the start position of the extent inside a larger file. 124 of the 128 bits in the GUID are truly unique for each extent, assuring that data can never be accidentally overwritten. If we were to put this into a theoretical perspective: as 124 bits allow for 2^{124} unique IDs, this system would allow generating one trillion new IDs per second for the next 674 thousand trillion years, only to illustrate that the storage system cannot run out of unique IDs.

In the Cache layer, each extent is split into 128 kB blocks identified by GUID, file block index and version number. For efficiency, when possible, very small updates are accumulated in RAM before they are saved to SSD. When appropriate, or when updates are committed, space is allocated on the cache disk for the new parts, for each 128 kB block independently, and then written. The gateway returns with a confirmation that the file is successfully secured. There is a list of block entries that will tell the Object Store which blocks need to be saved to storage disk and with what type of redundancy.

Note that erasure coding is not used in the Cache layer, but instead in the Object Store layer. Erasure coding in the Cache layer would make write and update a more complex and time-consuming task, requiring the system to collect pieces, then update, recalculate and distribute the new erasure-coded pieces. However, the data belonging to write and update operations is still mirrored to different caches for protection, when the write cache replication feature is enabled.

In the Object Store layer, each block that was scheduled for storage is sliced into a number of parts, according to the erasure coding redundancy level. For instance, for a 4+2 scheme, each slice will be 128 / 4 = 32 kB, referred to as the stripe unit size, plus two parts generated for redundancy, with same unit size. The Object Store then asks the cluster, using a multicast message, which nodes are available for storage. Each node in the cluster gives information about its own load and free space directly to the node responsible for the file, thus keeping multicasting to a minimum. The six pieces are distributed accordingly, considering:

- Geographical position
- Free space in the storage node
- Current workload
- Tier policies
- Outcome of a random generator (to avoid excessively repetitive patterns)

The data pieces are now distributed and the Object Store layer is going to store the chunks physically to disk on six different nodes. When the process is successful, the cached file is marked as “clean” and is kept in the cache but could be removed later if new blocks of data require the space.

To see potential risks in case of power failure during a file write, see section Data Protection and Power Loss.

Multi-threaded I/O

With vNAS, cloud storage and server virtualization, there is a need for high throughput and low latency for large files and multiple concurrent users. A Compuverde storage cluster can handle thousands of I/O requests simultaneously, in parallel, handled intelligently with byte-range file locking and a deep queue for waiting requests. To be more precise, per node:

- Parallel I/O operations: 16 + number of GB RAM
- I/O queue depth: 256 per client

The demand for I/O queues is usually much lower both because the traffic is shared across the cluster, and due to the existence of the Cache layer within each node.

Locks and Concurrency

Concurrent reading or writing of files is possible, even across platforms. For instance, when a client opens a file for reading, the client will be granted a shared lock so that multiple clients can read, not write, the same file on any gateway through any protocol. For writing, the client will request an exclusive lock to keep out other clients that may want to read or write in the same time. This can be done for a specific byte range so that multiple clients can simultaneously read and write to different parts of the same file. This is essential for OSs and virtual machines, as they often need to read and write to different parts of a file at the same time.

Example of simultaneous file writing and reading from different nodes, to the same file:

1. File read is initiated by two clients. The requests are redirected to the owner of the file
2. Shared lock is granted to the two clients and read is started in parallel
3. A third client requests an exclusive lock to write to the same file
4. The shared (read) locks cannot be reclaimed so the write will have to wait
5. When the reads are finished, exclusive lock is granted and the write can begin

If, instead, the reading clients only request locks for some part of the file, and the writing clients request to update another part of the same file, the write process would not have to wait.

Also supported are SMB opportunistic locks, or oplocks, and NFS delegations, so that clients can request files for caching locally. This allows for faster updates using less network traffic and improved response time. If the lock is possible (no other clients have the file open), the request is granted. When another client tries to access the file, the first client is notified so it can do appropriate actions, (flush and close the file), before the new client can proceed. If the first client for some reason cannot answer, then the opportunistic lock will be degraded, thus invalidating the local copy of the non-answering node, so that the new client can access the file.

System requirements

System requirements per node for use with products in regular, integrated mode:

	Minimum	*Recommended*
CPU	x86-64 (4 cores)	x86-64 (4+ cores)
RAM Memory	12 GB	32+ GB
Boot disk	20 GB	60+ GB
Cache disk	20+ GB SSD	100+ GB SSD / NVMe
Storage disk	SAS / SATA disks	SAS / SATA disks
Network	1 x Gigabit NIC	2 x 10 Gigabit (for separate back-end)
Number of nodes	4 nodes minimum	Scale out from 4 nodes
Network switch	Gigabit switch or better	2 x 10 Gigabit switches

Manageable switches with IGMP support; IGMP should be enabled on the private network. IGMP snooping should be enabled in all affected switches and IGMP query on one switch only.

Note: For separated mode, keep in mind that cache disks are only used for gateway nodes and storage disks only for object store nodes, so the requirements should be adjusted accordingly. In all-flash mode, there will be no cache disk, only SSDs for storage.

Test methods and verification

The Compuverde data center is used for demonstrations and to continuously test functionality. This is done both on very large and small Compuverde storage clusters to ensure compatibility, reliability, security and to verify that performance is not compromised by new features.

The main test setup at Compuverde consists of:

- 300 servers
- 7500 HDDs and SSDs
- 17 PB total capacity
- Power Usage Effectiveness: 1.05

Compatibility validation

Compuverde is continuously working towards higher levels of compatibility so that important features delivered by a given protocol can be fully utilized.

Test tools for testing compatibility include:

- Microsoft SMB Protocol Family Test Suite
- CITI PyNFS
- Libiscsi

Current compatibility according to test suites, as of January 2017:
(compared to Red Hat RHEL 7 / Cent OS and when using Samba 4.5.1 for SMB protocol)

- SMB 2.0 - 3.1.1 88 % (compared to 46 %)
- NFS 4.0 97 % (compared to 97 %)
- NFS 4.1 98 % (compared to 99 %)
- iSCSI 90 % (compared to 49 %)

For more details on supported features, see section on Supported protocols.

Performance

Performance is tested to verify and make sure that it stays ahead of the market and that it is not affected due to new features.

Test tools for performance include:

- SPC-1 / SPC-2
- SPECsfs 2008 / 2014
- Bonnie++
- Swift OpenStack Benchmark (ssbench)
- DiskSpd
- IOzone
- VDBench

Reliability

As the system software is strictly divided into layers (Protocol, Gateway with Cache and Object Store), changes and improvements in one layer are normally not affecting other parts of the software. Reliability is ensured as all parts are implemented by Compuverde in C/C++, all memory allocations are controlled and then checked against potential memory leaks. Continuous tests are performed and verifications are done for consistency and persistency.

Summary

With data storage demands rising exponentially, service providers and large enterprises are searching for ways to leverage customer data efficiently and safely while controlling spiraling energy expenses. Compuverde software-defined storage helps reducing hardware needs and capital expenditure by working with less expensive, low-performing storage units, forming them into high-performance storage clouds. The use of a symmetric architecture eliminates bottlenecks that are a costly part of traditional storage solutions. With no single points of failure or performance bottlenecks to protect, customers are free to choose less expensive off-the-shelf hardware.

- **Performance and reliability**: Data is stored in multiple copies and can be separated geographically among different locations within the cluster. A highly efficient fault recovery process allows for nearly complete redundancy.
- **Self-healing**: When a disk or storage node breaks down, the cluster self-heals by recreating the missing data, resulting only in a temporary capacity decrease.
- **Simplified deployment and maintenance**: The Compuverde software architecture avoids the need for special-purpose or central nodes, greatly simplifying deployment and maintenance.
- **Lower cost**: By leveraging clusters of standardized servers, Compuverde stores petabytes of accessible data at the lowest possible cost.

Technical specifications

System features

Storage modes	• Scale-out NAS (bare metal) • Hyper-converged • Hybrid Cloud • Metro Storage Cluster
Access Protocols	• SMB (1 / 2.0 / 2.1 / 3) • NFS (3 / 4.0 / 4.1) • iSCSI (with MPIO) • OpenStack Swift + Cinder • Amazon S3 • NNTP back end storage
All-Flash	Yes
Tiering	Yes
Multi-tenancy	Yes (multiple file systems)
Network	
Virtual IP	Yes
Multiple NICs	Yes
VLAN	Yes
Bonding / Link Aggregation	• Round Robin • Active/Backup • XOR • Broadcast • LACP • Adaptive Transmit Load Balancing • Adaptive Load Balancing
Rolling upgrade	Yes
Cache support	NVMe / SSD / RAM
Cache mode	• Read/Write • Write Only
Mirrored write cache	On / Off
RAM write cache	On / Off
Authentication	• Active Directory • LDAP • Kerberos KDC • NIS • Local database

Management features

Management tool	Yes (Windows based) • Performance • Configuration • Health • Alerts
Alarms	• SNMP • E-mail
Logging	• Yes, management tool • Remote syslog (rsyslog) • SNMP v2c
API	• REST API • Usage statistics
OpenStack Cinder	Yes

Cluster features

Scalability	Linear, by adding new nodes 100+ billion files (exabytes of data)
Elasticity	Runtime change of cluster size
Self-healing	Yes (Automatic)
Automatic detection	Node failure Disk failure Data inconsistency
Healing mode	Prioritized automatic repair
Availability	> 99.999 %
Data addressing	File system Block Objects
Encryption	Data at rest: AES 256 bit XTS, one key for each file system

Directory / File / Object features

File Policy	Yes (at folder level)
Filters	Pattern (ex: *.jpg) Age (days/weeks/months/years)
Actions	Change file coding Change tier Retention (remove files after set time) WORM (write once read many)
Snapshot Policy	Yes (at folder level)
Schedule	Automated: Every hour/day/week Manual
Snapshots retained	Up to 253 (circular overwriting when threshold is reached)
Quota Policy	Yes (at folder level)
Folder size limit	GB / TB / PB
File coding	Copies Erasure coding
Copies	3 or 5 copies
Erasure coding	2+1, 2+2, 3+1, 3+2, 4+1, 4+2, 5+1, 5+2, 6+1, 6+2, 8+1, 8+2

Index